Framework for Responding to Network Security Events (FRNSE)

Justin "J.D." Doak, Aaron Lovato, CTN-5

LANL network security analysts face a difficult challenge in monitoring the high volume of logs from network sensors around the Laboratory and taking action on potential security threats. The purpose of the FRNSE (pronounced fren-zee) project is to automate the process of monitoring log files and sensor output and to then respond in near real-time to potential security threats. FRNSE integrates sensors, policy, and responses in a centralized manner providing consistency in our approach to network security operations.

The FRNSE architecture (Fig. 1) consists of three main components: sensors, agents, and one or more servers. Each component plays a critical role in detecting and responding to network security threats. Sensors are typically network appliances (e.g., TippingPoint) that analyze packets crossing the section of the network where they are connected. Some of these appliances are in-line and have the ability to block network traffic. These tools are referred to as intrusion prevention systems (IPS). Other appliances (e.g., Snort) are not in-line and can only detect potentially malicious behavior. These tools are referred to as intrusion detection systems (IDS). In addition to looking at raw packets, certain sensors look at flow data, which summarizes packet transfers between two hosts with an established connection. Sensors are the first link in the FRNSE response pathway because they alert the system that something of interest may have occurred.

Agents act as an intermediary between the sensors and the server(s). Sensors may generate a large number of alerts; however, we may only be interested in a small subset of those alerts. An agent is configured to monitor the output of a sensor and only generates FRNSE alerts (alerts sent to the FRNSE server) when the sensor produces output that we have predetermined to be of interest. Agents formulate a valid FRNSE alert from the raw sensor output and send that message to the server(s) over a secure socket layer (SSL) connection. Alerts contain a category, a severity, and a certainty, among other fields. Category is the type of alert generated (e.g., ExploitAttempt), severity is how damaging an event is, and certainty is our confidence that the event for which we are generating an alert actually occurred (i.e., it is not a false positive). These three fields, when combined with a policy, determine how we respond to an alert, if at all. See Fig. 2 for an example of a FRNSE alert.

The FRNSE server(s) provides a centralized mechanism for integrating the various network security appliances and responding to their output in a consistent manner. When alerts are first received by the FRNSE server, they undergo correlation processing. Correlation uses previous alerts and any other pertinent data to modify fields in the alert, such as increasing or decreasing severity, or perform other actions. For example, if we have seen different categories of alerts generated by the same IP address in the last 24 hours, we increase the severity. After correlation, we respond according to policy. Available responses include switch blocks, firewall blocks, ticket posting, vulnerability registration, and e-mail notification. The results of those responses are obtained, and then the alert, along with the responses and

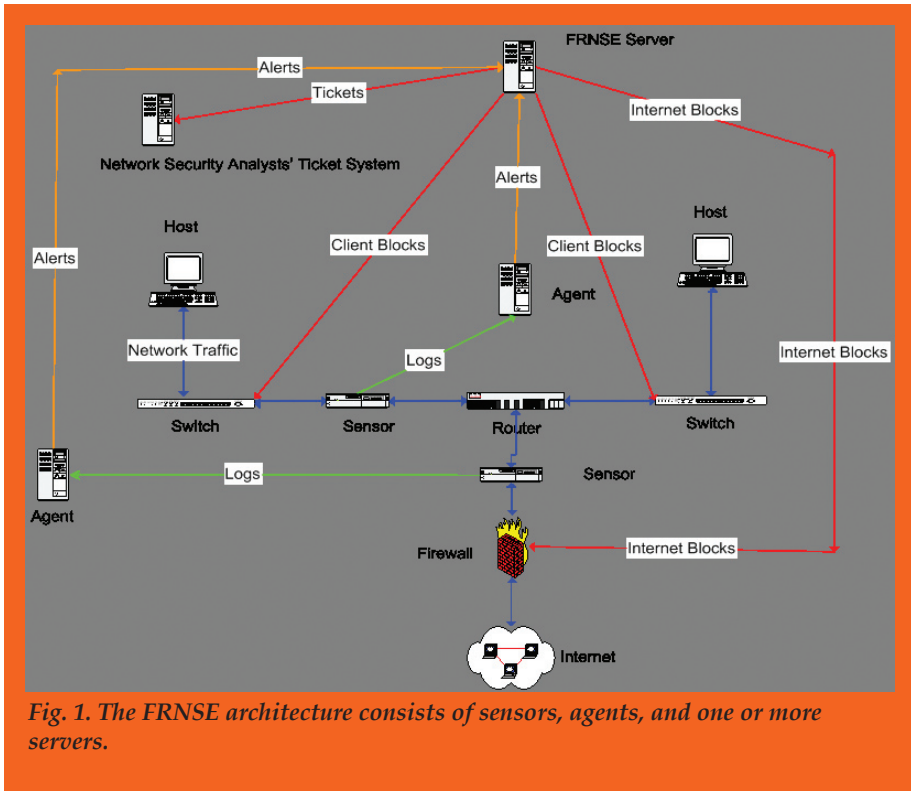

Fig. 1. The FRNSE architecture consists of sensors, agents, and one or more servers.

```
<alert>
  <category>ExploitAttempt</category>
  <severity>1.0</severity>
  <sensor>CFlow</sensor>
  <certainty>1.0</certainty>
  <taskmsg>srcip = 128.165.251.231 count = 11 start.date = 2007-11-28 start.ctime =
    15:03:06 end.ctime = 15:03:06 clock = 1993812</taskmsg>
  <srcip>128.165.251.231</srcip>
  <usermsg>Worm activity (RPC related) was detected on this host.  Please visit
    http://patchme.lanl.gov/</usermsg>
  <type>CFlow</type>
</alert>
```

Fig. 2. The XML-style format of a FRNSE alert.

results, is stored in a database. The FRNSE GUI (Fig. 3) is an interface built on the alert database to provide analysts with access to the raw alert data.

To illustrate the operation of FRNSE, let's follow the life of a FRNSE alert from creation at the sensor to a set of responses in the FRNSE server. Let's say that a rule for one of the Snort sensors is triggered by a particular network packet, generating an entry in a log file. The Snort Agent, which is monitoring that log file, determines that this is one of the Snort rules for which we generate FRNSE alerts. This alert is sent over SSL to the FRNSE server where it undergoes correlation. The severity is doubled in this correlation step due to historical alerts generated by this IP. The category, severity, and certainty of the alert are then compared with policy to determine how to respond. We respond by requesting a switch block and posting a ticket for the network security analysts.

FRNSE has been actively responding to network security events since November 2006. It has processed millions of alerts that then generated tens of thousands of responses. (Many alerts are archived, but not responded to.) All of this is performed in near real-time so that potential security threats are rapidly mitigated before spreading and causing more damage. Possible future work includes tuning sensors to detect more threats, developing agents for more sensors, adding new responses (e.g., e-mail blocking), and improving correlation.

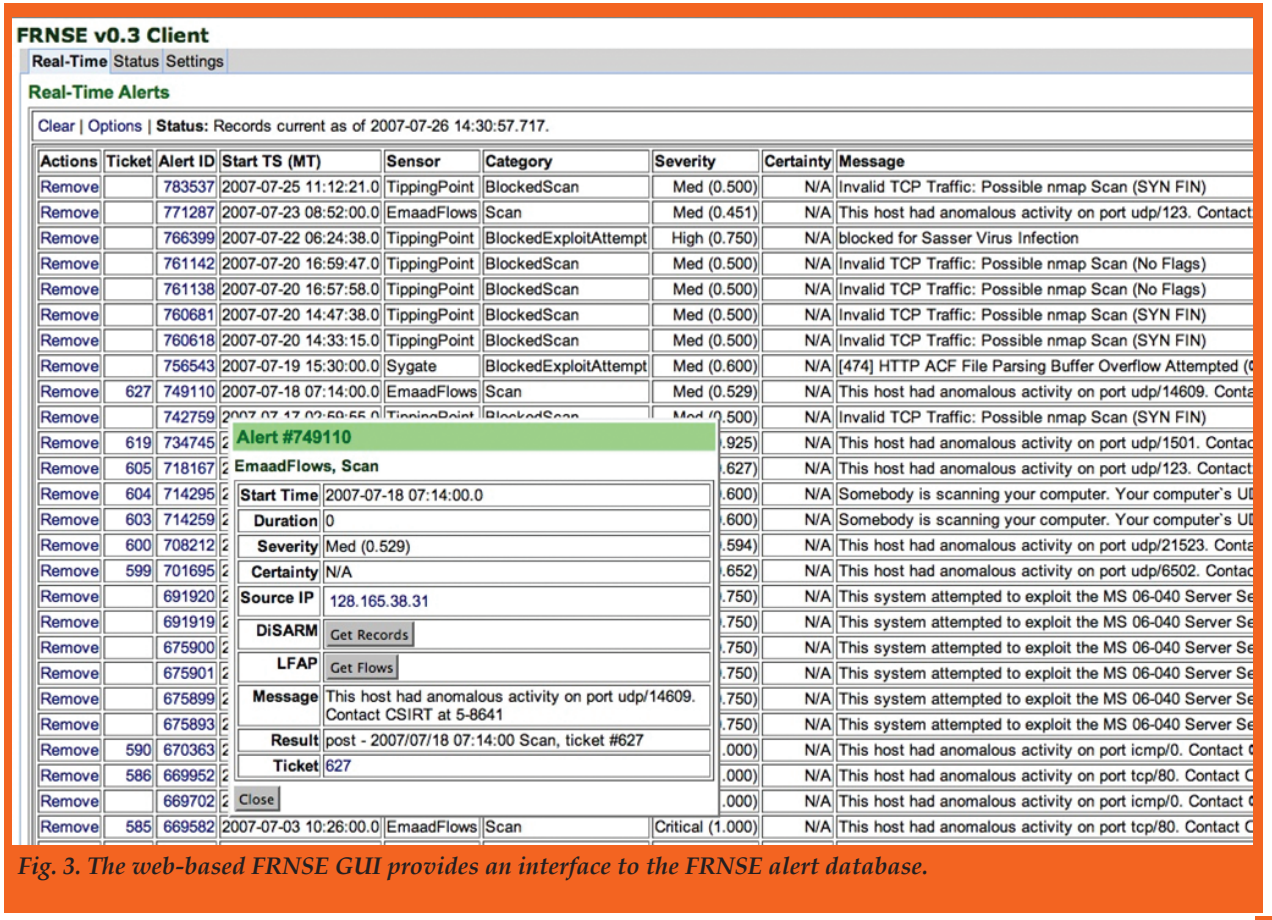

Fig. 3. The web-based FRNSE GUI provides an interface to the FRNSE alert database.

Funding Acknowledgements
- Computing, Telecommunications, and Networking Division (CTN) Classified and Unclassified Computer Security Programs

For further information contact Justin "J.D." Doak at jdoak@lanl.gov.